Collective Multi-Label Text Classification

Andrew McCallum
Department of Computer Science
University of Massachusetts
Amherst, MA 01002
mccallum@cs.umass.edu

Nadia Ghamrawi
Department of Computer Science
University of Massachusetts
Amherst, MA 01002
ghamrawi@cs.umass.edu

June, 2004

1 Abstract

Multi-label classification, the task of assigning one or more class labels to a document, arises in many domains. In the multi-label domain the categories need not be independent. As examples, a news article may be about multiple related topics and a medical journal article may pertain to multiple medical conditions. A common approach to multi-label classification is to train independent binary classifiers for each label, but this approach fails to exploit dependencies between labels. This paper explores conditional random field models for classifying documents that may have multiple labels. Single-label CRF models maintain features for word occurrence patterns; the models that this paper describes additionally maintain features corresponding to label co-occurrence patterns. These models outperform their independently trained single-label counterparts using several evaluation metrics on widely used corpora having varying characteristics. For example, even for sparsely multi-labeled corpora the models reduce subset classification error by as much as 30%. In addition, the models exhibit comparable F1-scores to the best performing classifiers under similar training conditions.

2 Introduction

Given a set of categories in a domain of text documents, a document may belong to multiple categories, and the categories need not be independent. For example, a news article may be about both oil and pollution, two topics that are related, specifically with respect to articles about the envinronment. Successful multi-label document classification, the task of assigning a document to one or more categories, is more complicated than the single-label classification task, since the categories may not be mutually exclusive. For instance, a news article about the effects of an economic recession on public health might belong to the topical categories HEART FAILURE, STROKE and RECESSION. The multi-label classification task arises in domains such as news article classification and medical diagnosis research. In such situations, text documents in different classes may be similar with respect to patterns in their features. In other multi-label domains, classes are not mutually exclusive, by definition. One such domain is semantic scene analysis [Boutell et al., 2003], which includes the task of automatically classifying images into categories according to descriptions of their scenes. The classes MEADOW and FOLIAGE are not mutualy exclusive, since a meadow scene is likely to be also a foliage scene (whereas STROKE and RECESSION are not inherently related).

*This work was funded in part by the Air Force Office of Scientific Research. The opinions expressed here are the authors and not necessarily those of AFOSR.

A common approach to automatic classification of documents belonging to more than one class is to independently train a binary classifier for each class, and classify a document into a category if the corresponding binary classifier scores above some threshold. Comparative experiments using 14 classifiers have been conducted [Yang, 1998], among which the classifier methods Widrow-Hoff, k-nearest-neighbor, neural networks and Linear Least Squares Fit Mapping had the best performance using the micro-averaged precision-recall breakeven point as a metric. Support vector machines are also powerful models for this task [Joachims, 1998]. Decision tree learning methods that exploit information gain [Chen and Ho, 2000] demonstrate comparable performance.

Another approach to multi-label document classification involves category ranking, in which each category receives some ranking in classifying a document x, and x is deemed to belong to the categories that rank above some threshold. For example [Schapire and Singer, 1999] develop a boosting algorithm that discovers a weak real-valued hypothesis over documents and labels that gives rise to such a ranking. The ranking model described in [Crammer and Singer, 2002] learns a prototype feature vector for each class given training data, so that the rank of a class is the angle between its prototype and d. For each category, the model in [Gao, et al., 2004] trains independent classifiers that may share some parameters, and ranks each classification according to a classification confidence measure. However, none of these approaches leverages information about label co-occurrences in determining multi-labelings.

An ideal model for multi-label classification would directly capture the dependencies between labels, where these approaches do not: knowing that with a high proability, a document belongs to the category IRAQ and has the term $energy$ increases the probability that the document belongs to the category OIL: the topics OIL and IRAQ may co-occur relatively frequently, and one might expect that documents having the word $energy$ would be about oil, but the probability that an article having the word $energy$ is about Iraq is lower.

This paper presents two new graphical multi-label models that capture the dependencies between pairs of labels and between labels and terms. In these models, edges correspond to features, and using the bag-of-words approach to natural language processing, nodes correspond to class labels and to terms. A feature is a function of a document and a class label in single-label classification. In the multi-label case a feature can be a function of a document and a set of class labels. The multi-label models developed in this paper maintain one feature for each term and label, capturing the co-occurence patterns between individual terms and labels. Additionally the first model, the Collective Multi-Labels classifier (CML), maintains features for each pair of labels, and the second model, the Collective Multi-Labels with Terms classifier (CMLT), maintains, for each term, a feature corresponding to each pair of labels.

Three multi-label corpora are used in experiments: Reuters-21578, OHSU-Med, and a small corpus of software usability reports. Each corpus has a different structure for its label taxonomy, different size and different noise level. The multi-label models presented outperformed the independent binary classifier approach using all three datasets. In particular, the models reduced error in subset classification by as much as 27%. Furthermore, in most experiments, the multi-label models resulted in higher per-label F1 scores, and sometimes reduced error in macro and micro averages by 9%. On benchmarked experiments with the Reuters-21578 corpus, the models resulted in comparable F1 macro- and micro- averages to that of the best reported results.

3 Three Models for Multi-Label Classification

Conditional models for text classification provide a rich framework for representing relationships between classes and features of a given domain. Furthermore, conditional models often outperform their generative counterparts in text classification.

Conditionally trained undirected graphical models, or conditional random fields, can be used to model

dependencies between terms and labels. These dependencies determine the conditional probability values on certain output nodes given the values of other input nodes by, given some clique parameterization, constraining the uniformity of the distribution. Maximum entropy is based on the principle that the distribution over variables corresponding to output nodes is uniform, barring these constraints. **make a stronger connection here, i.e., because the constraints are the expected values of the features, or clique parameterizations?** Therefore maximum entropy models can be interpreted as conditionally trained CRF models.

Maximum entropy solutions to natural language processing issues have shown to be competitive means of estimating probability distributions from data. The principle is appealing since it assumes no other dependencies apart from those explicitly captured by the graphical model. In fact, in several cases maximum entropy solutions outperform competitors. Maximum entropy principles have been employed in language modeling [Chen and Rosenfeld, 1999, Rosenfeld, 1994], and text segmentation [Beeferman, et al., 1999, McCallum, Frietag, and Pereira 2000]. Other uses of maximum entropy in natural language processing include named entity recognition [Chieu and Ng, 2002] with binary-valued features for frequently used words in the corpus, useful n-grams, and word suffixes, for example. Maximum entropy has also been applied to part-of-speech tagging using contextual features [Ratnaparkhi, 1996], identifying sentence boundaries given some features representing the context of the punctuation mark [Reynar and Ratnaparkhi, 1997], and determining the actions taken by a text parser using the context of an instance (whose features are contextual predicates) [Ratnaparkhi, 1997]. The maximum entropy approaches to the problem of document categorization presented in this paper use the bag-of-words model and employ features to capture co-occurrence patterns. The constraints are the expected values of these features.

3.1 Single-label Model

In maximum entropy classification of single-label documents, any real-valued function of the document x and class y, $f_i(x, y)$ can be treated as a feature. The trainer seeks to estimate the distribution of the class labels given x. Given training data $D = \{\langle \mathbf{x_1}, y_1\rangle, \langle \mathbf{x_2}, y_2\rangle, ..., \langle \mathbf{x_r}, y_r\rangle\}$ (having vocabulary V and set Y of classes), and some feature f_i, the estimated distribution $p(Y|\mathbf{x})$ must have the following property [Nigam, Lafferty, and McCallum, 1999]:

$$\frac{1}{D}\sum_{d=0}^{r} f_i(\mathbf{x_d}, y_d) = \frac{1}{D}\sum_{d=0}^{r}\sum_{y} P(y|\mathbf{x_d}) f_i(\mathbf{x_d}, y). \tag{1}$$

Thus in the case of single-label documents, the constraints are the expected values of the features computed using the training data. Given the features f_i, and λ_i, the parameters to be estimated, and the variable k enumerating the features, the distribution $p(y|\mathbf{x})$ is always of the parametric exponential form

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_k \lambda_k f_k(\mathbf{x}, y)\right), \tag{2}$$

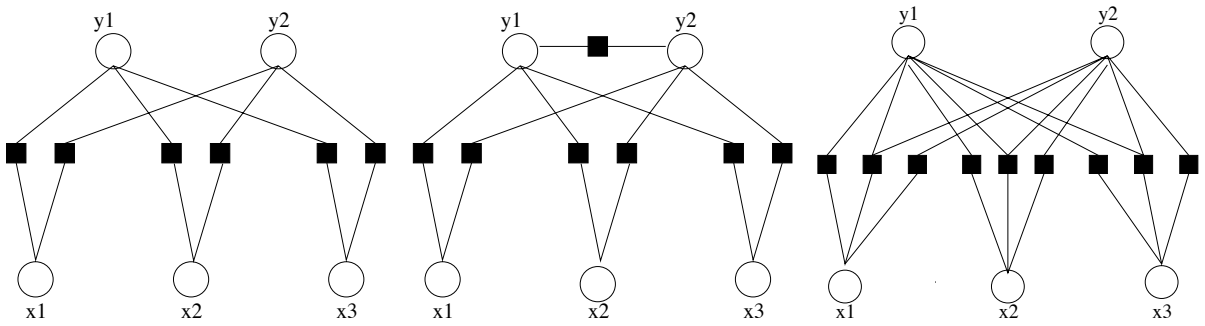
where

$$k \in \{\langle w_i, y_j\rangle : 1 \le i \le |V|, 1 \le j \le |Y|\},$$

and $Z(\mathbf{x})$ is the normalizing constant with respect to the features:

$$Z(\mathbf{x}) = \sum_y \exp\left(\sum_k \lambda_k f_k(\mathbf{x}, y)\right). \tag{3}$$

The distribution $p(y|\mathbf{x})$ is guaranteed to have this form, and furthermore the likelihood surface is convex [Berger, et al., 1996] (having one global maximum and no local maxima). Thus a hill climbing approach



(a) Single-label model. (b) CML model. (c) CMLT model.

Figure 1: Factor graphs representing the three maximum entropy models, where y is a label and x_i is a term.

to finding the maximum, starting from any initial exponential distribution, suffices to find the maximum entropy solution. BFGS [Byrd, et al., 1994] is a fast optimization method that can be used to find the global maximum of the likelihood function given the gradient.

Given D, the log likelihood of parameters Λ is

$$l(\Lambda|D) = \log g(\Lambda) \left(\prod_{d=1}^{r} p(y_d|\mathbf{x_d}) \right) = \sum_{d=1}^{r} \sum_{k} (\lambda_k f_k(\mathbf{x_d}, y_d) - \log Z_\Lambda(X)) - \sum_{k} \frac{\lambda_k^2}{2\sigma^2}, \tag{4}$$

where the last term is due to the Gaussian prior used to reduce overfitting. The maximum entropy learner attempts to find a Λ that maximizes the log likelihood iteratively. The gradient of the log likelihood function (with respect to Λ) at k is

$$\frac{\delta(\Lambda|D)}{\delta\lambda_k} = \sum_{d=1}^{r} \left(f_k(\mathbf{x_d}, y_d) - \sum_{y} f_k(\mathbf{x_d}, y) p(y|\mathbf{x_d}) \right) - \frac{\lambda_k}{\sigma^2}. \tag{5}$$

Solving for the pivotal λ requires an optimization algorithm, since this is not a closed form solution.

3.2 Accounting for Multiple Labels

In the CRF model in which each document has only one label, computing the conditional probability of each label involves a summation over features. If a document has multiple labels, then the conditional probability distribution that the multi-label model learns is over subsets of the label set Y, viewed as binary-valued functions of Y, which we will represent as a bit vector $\mathbf{y}$ of length $|Y|$. Let $D = \{\langle \mathbf{x_1}, \mathbf{y_1} \rangle, \langle \mathbf{x_2}, \mathbf{y_2} \rangle, ..., \langle \mathbf{x_r}, \mathbf{y_r} \rangle\}$ be the training data. In the most general form, given instance x_d

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(X)} \exp \left(\sum_{k} \lambda_k f_k(\mathbf{x}, \mathbf{y}) \right), \tag{6}$$

where

$$k \in \{\langle w_i, y_j \rangle : 1 \leq i \leq |V|, 1 \leq j \leq |Y|\}$$

and $Z(X)$ is the global normalizing constant.

3.3 Binary Model

The binary CRF model trains an independent binary classifier for each label y_b, using a particular partition of the training instances into positive and negative classes, resulting in a distribution p_b over the set of labels $\{+, -\}$. Then $p_b(y|\mathbf{x_d})$ is as in Equation 2, except that $y \in \{+, -\}$ and

$$k \in \{\langle w_i, y_j \rangle : 1 \leq i \leq |V|, 0 \leq j \leq 1\}.$$

However,

$$p(\mathbf{y}|\mathbf{x}) = \prod_b p(\mathbf{y}_b|\mathbf{x}), \tag{7}$$

where $\mathbf{y}_b$ is the value of label y_b in $\mathbf{y}$. As before, the binary model maintains a feature for each pair consisting of a word and a label, but the classifiers are completely independent.

3.3.1 CML Model

CML, like the single-label model, maintains a feature for each pair consisting of a label and a term. However, it additionally maintains features accounting for label co-occurrences. That is, for each document feature vector $\langle \mathbf{x}, \mathbf{y} \rangle$ and each distinct pair of labels y' and y'', there are four features according to the co-occurrence pattern, q, between those labels in Y:

q	feature
0	neither y' nor y'' labels $\mathbf{y}$
1	y' but not y'' labels $\mathbf{y}$
2	y'' but not y' labels $\mathbf{y}$
3	both y' and y'' label $\mathbf{y}$

Therefore in classifying documents in a domain of n labels, a document will have $4\binom{n}{2}$ label-pair features to consider, together with the label-term features. Experiments described represent these features with binary values. Thus, for example, if a training document has features $\mathbf{x}$ and label vector $\mathbf{y}$, then for $k' = \langle \text{WHEAT}, \text{GRAIN}, 2 \rangle$, $f_{k'}(\mathbf{x}, \mathbf{y})$ is 1 if the document is labeled GRAIN but not WHEAT, and 0 otherwise. With variables k and k' enumerating the two types of features, the probability distribution thus becomes

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\Lambda(\mathbf{x})} \exp\left(\sum_k \lambda_k f_k(\mathbf{x}, \mathbf{y}) + \sum_{k'} \lambda_{k'} f_{k'}(\mathbf{y})\right) \tag{8}$$

where

$$k \in \{\langle w_i, y_j \rangle : 1 \leq i \leq |V|, 1 \leq j \leq |Y|\},$$
$$k' \in \{\langle y_i, y_j, q \rangle : 1 \leq i, j \leq |Y|, 1 \leq q \leq 3\},$$

and

$$Z_\Lambda(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left(\sum_k \lambda_k f_k(\mathbf{x}, \mathbf{y}) + \sum_{k'} \lambda_{k'} f_{k'}(\mathbf{y})\right).$$

The CML log likelihood of the parameters Λ is similar to that in the singly-labeled maximum entropy model:

$$l(\Lambda|D) = \sum_{d=1}^{r} \left(\sum_k \lambda_k f_k(\mathbf{x_d}, \mathbf{y_d}) + \sum_{k'} \lambda_{k'} f_{k'}(\mathbf{y_d}) - \log Z_\Lambda(\mathbf{x_d})\right) - \sum_k \frac{\lambda_k^2}{2\sigma^2} - \sum_{k'} \frac{\lambda_{k'}^2}{2\sigma^2}. \tag{9}$$

Likewise, the gradient of the log likelihood function at k is

$$\frac{\delta(\Lambda|D)}{\delta\lambda_k} = \sum_{d=1}^{r}\left(f_k(\mathbf{x_d},\mathbf{y_d}) - \sum_{\mathbf{y_d}} f_k(\mathbf{x_d},\mathbf{y_d})p(\mathbf{y_d}|\mathbf{x_d})\right) - \frac{\lambda_k}{\sigma^2}, \tag{10}$$

but at k' it is

$$\frac{\delta(\Lambda|D)}{\delta\lambda_{k'}} = \sum_{d=1}^{r}\left(f_{k'}(\mathbf{y_d}) - \sum_{\mathbf{y_d}} f_{k'}(\mathbf{y})p(\mathbf{y}|\mathbf{x_d})\right) - \frac{\lambda_{k'}}{\sigma^2}. \tag{11}$$

CML captures the flavor of label co-occurrences in the corpus independent of the feature values of the document. Effectively, for each label set, it adds a bias that varies proportionally to the label set frequency in training data. Figure 1(b) represents this model.

3.3.2 CMLT Model

Label co-occurrenes in a document labeling are not independent of the word features in the document. For instance, a document belonging to the categories RICE and SOYBEAN might have increased likelihood of being automatically correctly classified if the document has the term *cooking*, while *cooking* reduces the belief that the document belongs to the category ALTERNATIVE FUELS. Similarly, the presence of the term *cars* increases the likelihood that the document belongs to the category ALTERNATIVE FUELS but decreases the likelihood that the document belongs to SOYBEAN. In that spirit, as the factor graph in 1(c) reflects, the edge potential between labels - that is, the label pair features - is not independent of the terms. The CMLT model maintains parameters that correspond to features for each $\langle term, label_1, label_2\rangle$ triplet, capturing parameter values for $\langle cooking,$ RICE, SOYBEAN$\rangle$, and $\langle cooking,$ ALTERNATIVE FUEL, SOYBEAN$\rangle$, for example.

Then CMLT defines feature parameters over the labels and words as in CML,

$$k \in \{\langle w_i, y_j\rangle : 1 \le i \le |V|, 1 \le j \le |Y|\},$$

but, unlike CML, also defines parameters over pairs of labels and words,

$$k' \in \{\langle w_i, y_{j_1}, y_{j_2}\rangle : 1 \le i \le |V|, 1 \le j_1, j_2 \le |Y|\},$$

for a total of $O(n^2|V|)$ parameters for n labels. The corresponding conditional probability distribution that CMLT learns is

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\Lambda(\mathbf{x})}\exp\left(\sum_k \lambda_k f_k(\mathbf{x},\mathbf{y}) + \sum_{k'} \lambda_{k'} f_{k'}(\mathbf{x},\mathbf{y})\right). \tag{12}$$

The gradients of the log likelihood at k and at k' are the same as that of Equations 10 and 11, respectively, except that k' enumerates different features (so $p(\mathbf{y}|x)$ is as defined in Equation 12).

Note that unlike CML, CMLT does not capture the co-occurrence relationship q between each pair of labels. In CMLT, the k' feature is non-zero for a given training document only if both labels label the training document which has at least one occurrence of the term.

Factor graphs can be used to represent the three types of models (Figure 1). Factor graphs, rather than graphical models, graphically illustrate which parameteration of cliques each model considers in learning the conditional distributions [Bunescu and Mooney, 2004, Kschischang, 2001]. The features correspond to factors in the factor graphs. In the factor graphs in Fig. 1(b), for example, the features involve pairs of labels as well as term-label pairs; in Fig. 1(c), for each term, there is a feature for each label and a feature for each pair of labels.

Learning model parameters in these models is exactly the same as in the single-label model: BFGS is used to find the optimal parameters given the gradient of the log-likelihood functions. Note that neither multi-label maximum entropy model assumes that the label taxonomy has any complex structure (for example, hierarchical) – each simply accounts for label co-occurrences.

Input: set D of training documents d having feature vector $\mathbf{x}$ and labels $\mathbf{y}$, and set S of combinations of labels
Output: classifier with learned parameters Λ for computing $p(\mathbf{y}|\mathbf{x})$ of unseen feature vector $\mathbf{x}$.

1. Initialize maximizable trainer:

   compute constraints C from feature vectors x and label(s) $\mathbf{y}$ as expected values of features. That is,

   $$C_k = \sum_{<\mathbf{x},\mathbf{y}>} f_k(\mathbf{x},\mathbf{y})$$

   and

   $$C_{k'} = \sum_{<\mathbf{x},\mathbf{y}>} f_{k'}(\mathbf{x},\mathbf{y}),$$

   where k' is as defined for the given multi-label classifier.

2. Initialize a maximizer function (l-BFGS)

3. Repeat for maximum iterations or until converged:

   use maximizer to maximize the log likelihood of Λ, producing Λ':

   a. with current parameters Λ, compute the gradient of the log likelihood: For each training document having features $\mathbf{x}$, use Λ to score all $\mathbf{y} \in S$, resulting in $p(\mathbf{y}|\mathbf{x})$. That is, for each k compute

      $$-\sum_{\mathbf{y}} f_k(\mathbf{x},\mathbf{y}) p(\mathbf{y}|\mathbf{x}),$$

      and analogously compute the gradient point for each k'.

   b. add C to the gradient

   c. use the result with l-BFGS to compute Λ' that maximizes log likelihood, and set Λ to Λ'.

4. Output resulting classifier with learned parameters Λ.

Figure 2: The learning algorithm.

Figure 3.3.2 summarizes the learning algorithm. CMLT models maintain a parameter for each feature and pair of labels that occur in the training data. Given a vocabulary of v terms, s distinct subsets of k labels, for each training document, the CMLT training algorithm has complexity $O(vk^2s)$ in worst case, and the model requires space $O(vk^2 + s)$. CML, which accounts for label co-occurrences irrespective of features, has training complexity $O(vks + vk^2)$ per document and requires space $O(vk + s)$. In contrast the binary classifiers require time and space asymptotically the same as that of CML, but the independent binary classifier technique is faster than the multi-label models in most experiments.

4 Inference

CML and CMLT can learn a probability distribution over select subsets of a given label set. For instance, the models may learn a distribution over *subsets* of the 10 most common training labels in a corpus. For example, this may be the distribution over all possible combinations of the 10 labels, or it may be a distribution over only combinations of labels that occur in the training data. CML and CMLT, as inference tools for classification, can provide a means for attributing the best subset of labels to an instance, regardless of the presence of the combination in training data. There are three alternatives for inference.

4.1 Exact inference

Exact inference requires learning a probability distribution over all subsets of labels. This method is intuitively appealing because it is easy to explain and informative in that it offers a probability score for each combination of labels, regardless of the combinations presence in the training data. There are 2^n such subsets for n labels, however, so the problem is tractable for 3-12 classes, but it becomes intractable when there are more than about 12-15 classes.

4.2 Supported inference

CML and CMLT efficiently learn the distribution over only the label combinations that occur in training data (that is, the combinations that are supported by the training data). For the top 10 classes in Reuters-21578, of the 2545 test instances, 16 belong to combinations of categories that do not occur in training data, so these instances must have incorrect subset classification. In general as there are more classes relative to the corpus size, the likelihood of rare combinations increases. For the entire ModApte split, the error due to this scheme is more significant: there are 117 of 2998 test instances with combinations of labels that do not occur in training data. If CML and CMLT use this method of inference, then comparing techniques requires excluding these test instances in computing the performance metrics. Thus when there are few classes and few such outliers, or when such rare combinations are believed to only introduce noise, then supported inference is a reasonable solution.

4.3 Binary pruning

Binary pruning is one compromise of the supported and exact inference methods. Instead of purely exhaustive inference, a solution to the inference problem is to first train an independent maximum entropy binary classifier for each label. For each document, each classifier then has a probability score for the respective class, and exhaustive inference can be performed using the labels having probability scores above a certain threshold. The advantage of this technique is that it is possible to correctly classify test documents whose actual combinations do not occur in the training data, but it is more feasible than purely exact inference. Furthermore, by offering a hierarchical solution selecting only the best scoring classes for each document, this approach reduces overfitting of the model parameters to the training data. In general, using a threshold

value that results in competitive performance as the supported inference method with the same vocabulary, binary pruning requires less training time than supported inference.

5 Experiments

Experiments with the multi-label classifiers were done using three multi-label corpora: Reuters-21578, OHSU-Med, and a small corpus of usability reports, each of which Aptima, Inc.* culled from the Web and manually classified. The corpora differed in the noise level and length of documents as well as in the label taxonomy and distributions of the labels.

5.1 Document Reprsentation

Document term vectors can be represented several ways. With the exception of the k' features of CML, term counts were used for experiments. That is, given document x with label vector $\mathbf{y}$,

$$f_{w_k,y_k}(x,\mathbf{y}) = N(\mathbf{x}, w_k),$$

if the document was labeled with with y_k, and

$$f_{w_k,y_k}(x,\mathbf{y}) = 0,$$

otherwise, where for a given document, $N(\mathbf{x}, w_k)$ is the number of occurrences of term w_k in $\mathbf{x}$. In some experiments, terms occurring fewer than k times in all training documents were "cut" from consideration. Additionally, terms that are "uninformative" are excluded from the vocabulary. That is, some terms may occur frequently in documents of a particular class but less frequently in documents of other classes. Such terms are very informative, so they should have a higher weight than their counterparts. The"mutual information" of each term is calculated based on the subset of classes it occurs in, similarly to the information gain definition in [Mitchell, 1997]:

$$MI(w,D) = H(D) - \sum_{v \in Values} \frac{|D_{w,v}|}{|D|} H(D_{w,v}),$$

where $H(S)$ is the entropy of set S, $Values$ is the set of values that term w can take on, D is the document corpus, $D_{w,v}$ is the set of documents having value v for term w. Entropy signifies the impurity of a set of instances:

$$H(S) = -\sum_{y=1}^{|Y|} p_y \log p_y,$$

where p_y is the proportion of instances in S labeled y. The entropy in the set is highest when the distribution over classes is uniform. Then terms can be ranked according to their MI value. In experiments described in this paper, mutual information was used to rank terms and select the most informative ones. This resulted in improved test set performance for all classifiers.

5.2 Label Taxonomies

Labels of text documents in a multi-label corpus may be related in systematic ways. That is, the label taxonomy may have a specific structure. Three major types of label taxonomies are hierarchical, multi-dimensional, and simple.

Simple The multiple labels of a given document are not constrained by any structure imposed by the taxonomy, such as categories and sub-categories of labels. For instance, a document might have topics WHEAT as well as CORN.

Hierarchical A document labeling may belong to classes and subclasses. For instance, a document about grain and wheat may be labeled GRAIN as well as WHEAT, while another document about tomatoes and corn might be labeled CORN, TOMATO and VEGETABLES.

Multi-dimensional Each dimension of the label taxonomy consists of several possible labels. For instance, a classification taxonomy may require all documents to have a "topic" label for which documents may be labeled CORN, as well as a "place" label such as ILLINOIS, NORTH AMERICA.

Note that hierarchical and multi-dimensional label taxonomies can be generalized to simple taxonomies. For instance, a document in class GRAIN and subclass WHEAT could be considered part of the single class GRAIN-WHEAT. The multi-label models described in this paper best represent the simple taxonomy.

5.3 Corpora

5.3.1 Reuters-21578

Widely used Reuters-21578 experiments on the entire corpus typically employ the ModApte version of the corpus, in which all labeled documents that occur before April 8, 1987 are treated as training documents, and all labeled documents after this date are used in testing. Furthermore, it includes only classes which have at least one training and one testing document, for a total of 90 classes and 94% of labeled documents. There are 10,702 total documents, 7,704 of which are training documents.

The Reuters-21578 topical dataset has a simple label taxonomy. Each document belongs to one or more topical classes. Roughly 8.7% of the Reuters-21578 documents have more than one topic label. The Reuters experiments used subsets of the dataset which included documents that were "tightly labeled". That is, ideal classes, say APPLE, ORANGE, and CITRUS, have the properties that APPLE and ORANGE co-occur frequently, and CITRUS either occurs infrequently with the former classes, or co-occurs frequently with either APPLE or ORANGE, but not both. Table 10 depicts the distribution of documents corresponding to one such subset of classes, hereafter known as *ReutSM*.

For ReutSM, each of the 15 combinations of the five labels that occurs in the training set also occurs in the test set The label SHIP exclusively labels a document 87% of the time, while 10% of the time it co-occurs with GRAIN, the largest category labeling 61% of instances and belonging to 10 of the 15 distinct combinations of labels occurring in the training data. Furthermore 99% of WHEAT instances are also GRAIN instances.

Other Reuters experiments presented in this paper used only the documents belonging to the 10 largest classes (*Reut10*), which included about 84% of the documents in the ModApte split and formed 39 distinct combinations of labels occurring in the training data. Table 1 depicts the distribution of labels in Reut10. Notably 98% of ACQ instances belong to no other category, and similarly for EARN. Among MONEY-FX instances, 84% are INTEREST instances and 16% are TRADE instances. For most classifiers, reports of *ReutAll* experiments, using all 90 classes and corresponding documents of the ModApte split, having 372 distinct combinations of labels in training, is intractable. Although the multi-label models required several days to train, and CMLT classifiers required considerable memory, feature selection using information gain and cutting features facilitated model training and improved their performance. Table 2 depicts the distribution of multi-label cardinalities in the ReutAll test set, together with the label classification error rate of the binary classifiers. The fact that the error rate increases as the multi-label cardinality increases suggests that the labels are not independent and furthermore that it is advantageous to leverage the label co-occurences.

label	count	multi-label count
grain	582	469
wheat	283	282
corn	237	237
money-fx	717	218
acq	2369	50
crude	578	120
earn	3964	34
interest	478	188
trade	486	60
ship	286	120

Table 1: *Distribution of documents in the Reuters-21578 corpus over categories of Reut10, where the first column is the count of documents belonging to the respective category, and the second column is the number of documents belonging to at least one other category. For example,* WHEAT *and* CORN *almost always co-occur with some other label, but* ACQ, EARN *and* TRADE *hardly ever do.*

number of labels	1	2	3	4	5	6	7-14
number of documents	2561	308	64	32	14	6	13
binary model error	0.142%	0.641%	1.46%	1.98%	1.85%	3.33%	5.83%

Table 2: *Histogram of ReutAll test set combinations of labels by combination cardinality, and the binary model label mis-classification rate (that is, the fraction of times a category was incorrectly classified for each label cardinality). Note that as the cardinality of a multi-labeling increases, the binary models are more likely to incorrectly attribute (or fail to attribute) a label to a document. This suggests that it is potentially advantageous to consider label co-occurrences in classifying a document.*

5.3.2 OHSU-Med

The OHSU-Med corpus [Hersh, et al., 1994], commonly used in text classification, is a collection of titles and abstracts of medical research journal articles from 1989-1991. There are over 14,000 unique categories, and each article belongs to an average of 13 categories. The Medline label taxonomy is hierarchical, where labels may have multiple sub-categories (so a multi-labeling corresponding to one document need not have only a subcategory and all of the ancestor categories). OHSU-Med experiments in this paper use the 1991 documents as training data and 1990 documents in testing. Of these, only those documents that are labeled with some subcategory of the category 'Heart Disease' (*HD*) are used, for which there are 119 subcategories. The *HD-small* documents used in Heart Disease experiments are labeled by the 40 categories which label between 15 and 74 training documents, forming 106 combinations of labels in the training data. Similarly the *HD-big* experiments include documents labeled by the 16 categories which label 75 or more training documents.

There are 126 distinct combinations of labels in the training data. Each instance is represented as a feature vector of the terms in the title concatenated with the abstract, if any. The categories correspond to characterizations of the relevant heart conditions, such as "Heart Aneyurism" and "Myocarditis".

Compared to the ReutersAll corpus, HD is noisier and documents belong to more categories on average. Moreover the heart disease collection of HD documents is more specialized than the Reuters collections of general news documents, so HD documents that do not have all common labels are more likely to be similar than such Reuters documents.

5.3.3 Usability

The third corpus, *Usability*, used in these experiments is a collection of software usability documents[1]. The corpus has a two-dimensional label taxonomy with dimensions Interface Aspect and Interface Element. Each dimension has three categories, and each document must belong to one category from each dimension. Thus in the simple taxonomy representation, there are nine classes, and each document has exactly two labels. The corpus consists of 750 usability documents gathered from the Web and manually classified by experts in the usability domain. Documents average 32 words in length. Table 3 depicts the distribution of usability reports across classes.

	Interface Element			
Interface Aspect	Control	Information Space	Application Component	Total
Presentation	33	54	11	98
Representation	185	30	1	216
Other	37	248	149	434
Total	255	332	161	748

Table 3: *Distribution of documents in the usability report training data. Cell* (i, j) *indicates the number of documents labeled with both* i *and* j*. Observe that the class* REPRESENTATION-APPLICATION COMPONENT *is small compared to* REPRESENTATION-CONTROL*, for example.*

Interface Element	Interface Aspect	Document
Control	Presentation	The button for the category that I'm search doesn't look very different from the other buttons.
Information Space	Other	I don't know how to cancel a search.
Control	Representation	I don't have any context for the search results. I don't know what to expect if I click the link for one of the users in the search results.

Table 4: *Three exemplary usability reports and their classifications.*

The former two corpora have simple label taxonomies and are large. The usability corpus has a more intricate two-dimensional label taxonomy, and it is smaller, noisier, and has shorter documents than both corpora. Furthermore, some categories are several factors larger than others. These characteristics make the usability corpus an interesting corpus for use in comparing classifiers.

5.4 Results

In all experiments, both multi-label models, as well as independent maximum entropy binary classifiers, are trained and tested using the same training and testing data. The results are compared using two metrics: F1-score micro-average and macro-average, and subset accuracy. The F1 score for binary classifier for label y is the harmonic mean of the classifier precision and recall. A multi-label classifier's F1-score for y is similarly defined.

Given an instance with labels $\mathbf{y}$, the set of binary classifiers have correct *subset* classification for the instance if each binary classifier corresponding to a label y gives a positive classification for y if and only if y is positive for $\mathbf{y}$, while the multi-label classifiers must classify the document $\mathbf{y}$. The classifier's subset accuracy is the proportion of test instances with correct subset classification.

[1] see Web source

The macro-average of F1 scores is the mean of the F1-scores of all the labels, thus attributing equal weights to the label F1 scores. The micro-average is the F1-score obtained from the summation of contingency matrices [Yang, 1998] for all binary classifiers. Thus the micro-average metric gives equal weight to all classifications, so that F1 scores of larger classes influence the metric more than F1 scores of smaller classes. Where the number of classes is large, comparing micro-average and macro-average of F1 scores for all class labels facilitates evaluation of the performance of multi-label, multi-class classifiers [Yang, 1998].

5.4.1 Reuters-21578

Parameters that influence performance of the classifiers include proportion of features selected, feature cut size, Gaussian prior variance of the classifiers, and in the case of binary pruning, the threshold for the binary classifiers. The binary pruning techniques are most sensitive to the threshold parameter within a certain range. In general, the binary pruning technique resulted in 0.7% higher F1 micro-average and 29% higher macro-average than supported inference using the optimal threshold, barring feature proportion. This suggests that binary pruning improves performance of smaller classes. Lower thresholds increase classification time but tend to improve performance. Higher thresholds reduce the sensitivity of the classifier to varying feature proportions.

The best published micro-average of F1 scores of classifiers on the ReutAll corpus is 0.88 using support vector machines [Yang, 2002]. With thresholds of 0.7, CML and CMLT achieve 0.87 F1 micro-average. Table 7 depicts these results, as well as results comparing the two inference methods on ReutAll. Performance of supported inference experiments are more sensitive to changes in proportion of features used, but they are more memory-intensive and require more time to run than binary pruning. Binary pruning performs slightly better than supported inference in these experiments, but performance remained the same for 70% or more features at most probability thresholds.

Table 9 depicts the error over multiple trials in classifying random test-train partitions of ReutAll, given that 70% of data is used in training.

For ReutSM and Reut10 experiments, multi-label classifiers yielded higher F1 scores than the corresponding binary classifier for most labels (Table 5). For ReutSM, CML reduced subset classification error by 9% and CMLT reduced it by 19%. For the larger and more complex Reut10, CML reduced error by 28% and CMLT reduced subset classification error by 22%. CMLT is more subject to overfitting, hence when there are many classes that are not as richly multi-labeled, the improvement subset classification is greater for CML than CMLT.

Compared to the binary classifiers, the multi-label models yielded 1% poorer recall for the label SHIP, however. Note that compared to other multi-labelings, very few SHIP documents belong to another class, and the exceptions almost always belong to GRAIN only. Hence upon misclassification, the multi-label classifiers most often mistake SHIP instances for, exclusively, GRAIN instances. The binary classifiers do not have this bias, since they do not consider the label co-occurrences. However, 96% of COFFEE instances are only about COFFEE, for which the multi-label classifiers had 2% better F1 than the binary classifiers. The label CORN occurs frequently with both WHEAT and GRAIN, which are larger classes. The multi-label classifiers therefore tend to believe CORN instances belong exclusively to WHEAT or to GRAIN.

Note that the multi-label classifiers have higher subset accuracy than the binary classifier. This supports the belief that the classes are not semantically independent.

Table 11 summarizes the distribution of documents with respect to GRAIN co-occurrence patterns with other labels in CML, together with the factor values.

ReutSM			
	Binary	CML	CMLT
Supported			
grain	0.953	0.973	0.976
wheat	0.853	0.861	0.868
corn	0.860	0.852	0.832
coffee	0.982	0.982	0.982
ship	0.966	0.955	0.944
macro-F1	0.923	0.925	0.920
micro-F1	0.928	0.932	0.930
subset accuracy	0.837	0.845	0.853
Binary Pruning			
macro-F1	0.923	0.925	0.925
micro-F1	0.928	0.932	0.932
subset accuracy	0.837	0.849	0.849

Table 5: *Performance of CML and CMLT, using supported inference and binary pruning at threshold 0.5, on ReutSM, using the most informative 60% of features. With few classes, performance of the multi-label models is slightly better than the performance of the binary model, but reduction in error is significant. Performance of CML and CMLT was identical using binary pruning, and binary pruning resulted in improvemed performance for CMLT (compared to supported inference).*

5.4.2 OHSU-Med

Subsets of the OHSU-Med corpus tend to be noisier than Reuters-21578. The intuitive reason is that OHSU-Med data refer to specialized topics within specific disciplines so that articles are likely to have similar word frequencies even if they do not have the same label (such as 'lymphocytes', 'postoperative', and 'ventricular'). The Retuers news articles, on the other hand, span a more general collection of topics, so that documents which have different labels are likely to have fewer common features. Table 13 depicts the performance of the three maximum entropy techniques on the heart disease subsets of OHSU-Med.

Table 13 depicts the performance of the three techniques on HD-small and HD-big using two inference methods. Like Reuters, OHSU-Med has a simple label taxonomy, but the subset accuracy of the multi-label classifiers is about 25% better than the subset accuracy of the binary classifiers using the supported classes. This sigificant improvement in performance by leveraging label co-occurrence patterns supports the earlier conjecture that the Heart Disease labels are semantically dependent even moreso than are the Reuters-21578 labels. The macro and micro F1 scores suggest that CMLT performance is more sensitive to the sizes of the classes.

5.4.3 Usability

As with Reuters-21578 and OHSU-Med experiments, usability documents are represented as term frequency vectors. Results are averaged over several trials, where for each trial the corpus is randomly partitioned, using 75% of documents for training and 25% for testing. Both multi-label models, as well as the independent binary classifiers, are trained and tested using this partition.

For Usability, the binary classifier resulted in 29% subset accuracy while the multi-label classifiers had subset accuracies 43-44% (Table 16). That the multi-label classifiers performed considerably better than the binary classifiers suggests that the classes are not independent. Note that the factor value for

Reut10			
	Binary	**CML**	**CMLT**
Supported			
macro-F1	0.867	0.871	0.865
micro-F1	0.935	0.938	0.936
subset accuracy	0.888	0.909	0.906
Binary pruning			
macro-F1	0.867	0.879	0.878
micro-F1	0.935	0.941	0.941
subset accuracy	0.888	0.907	0.906

Table 6: *Performance of the two techniques, supported classes and binary pruning, threshold 0.5, using the best 60% of features. CML and CMLT achieved greater error reduction, with respect to binary classifiers, for Reut10 than for ReutSM. Binary pruning proved advantageous.*

ReutAll			
Binary pruning, 40% features			
macro-F1	0.438	0.458	0.458
micro-F1	0.863	0.870	0.871
subset accuracy	0.800	0.832	0.832
classification time (ms)		78	114
Supported, 40% features			
macro-F1	0.438	0.448	0.384
micro-F1	0.863	0.866	0.843
subset accuracy	0.800	0.833	0.814
classsification time (ms)		48	78
Binary pruning, 70% features			
macro-F1	0.439	0.457	0.457
micro-F1	0.864	0.870	0.870
subset accuracy	0.800	0.832	0.832

Table 7: *Performance of the three maximum entropy techniques on ReutAll, using threshold 0.7 for binary pruning. Using 70% of features the results of binary pruning are comparable to best reported results. Also note that binary pruning achieves slightly better performance than the supported inference method.*

ReutAll, pruned test instances			
Binary pruning, 40% features			
macro-F1	0.247	0.221	0.221
micro-F1	0.492	0.444	0.444
Supported, 40% features			
macro-F1	0.470	0.259	0.180
micro-F1	0.899	0.512	0.427

Table 8: *Performance of the three maximum entropy techniques on the ReutAll pruned instances, using threshold 0.7 for binary pruning. In the case of binary pruning, binary classification results in better performance on pruned instances (instances whose actual combinations included classes for which the binary classifier attributed probability less than 0.7) than the CML and CMLT models.*

ReutAll			
	Binary	CML	CMLT
	macro-average		
mean	0.4427	0.4616	0.4616
std. dev.	0.00525	0.00553	0.00553
	micro-average		
mean	0.8747	0.8835	0.8835
std. dev.	0.00124	0.00091	0.00090
	subset accuracy		
mean	0.8096	0.8439	0.8439
std. dev.	0.00125	0.00103	0.00109

Table 9: *Mean micro- and macro-averages and subset accuracy of the binary pruning CML and CMLT classifiers over several trials, using 50% of features at threshold 0.7. Each trial uses 70% of ReutAll in training, the other 30% in testing, roughly the same proportions as the ModApte split. These results suggest that even with random test-train splits of the corpus, CML and CMLT outperform the independent binary classifiers.*

subset	count	subset	count
grain	83	wheat	1
grain,wheat	144	grain,corn	120
wheat,corn	2	grain,wheat,corn	54
coffee	104	grain,wheat,coffee	3
ship	166	grain,ship	19
grain,wheat,ship	5	grain,corn,ship	2
grain,wheat,corn,ship	2	coffee,ship	3

Table 10: *Distribution of subsets of labels in ReutSM training data. Note that* WHEAT *hardly occurs alone, and* GRAIN, WHEAT *and* CORN *together form a cluster of labels which co-occur frequently but* SHIP *and* COFFEE *form clusters that are mostly independent of the others.* COFFEE *is especially independent of the other clusters.*

	not grain and not x		grain and not x		not grain and x		grain and x	
wheat	273	0.0	102	0.0	3	0.000282709	209	0.052517
corn	274	0.049274	2	-0.020478	154	0.041982	179	0.171887
coffee	167	0.0	329	0.061735	107	0.0	4	0.0
ship	107	0.272986	405	-0.002133	169	0.200047	28	0.065969

Table 11: *Count of documents for each label-grain co-occurrence pattern for each label, together with the corresponding factor value learned by CML using ReutSM. These parameters suggest that in classifying an instance, for example, the score of a combination of labels that does not have the label* GRAIN *will be strongly influenced by the parameter corresponding to the presence or absence of* SHIP*, but the score of a combination that does have the label* GRAIN *will hardly be affected by the parameter corresponding to* $\langle$GRAIN, notSHIP$\rangle$.

smallest clique potentials			largest clique potentials		
labels	term	value	labels	term	value
grain,ship	blah	-0.3840	grain,corn	paris	0.1736
grain,wheat	corn	-0.2691	wheat,corn	grain	0.1775
wheat,corn	sold	-0.2169	wheat,corn	barley	0.1806
grain,wheat	barley	-0.2029	wheat,corn	bushels	0.1819
wheat,corn	blah	-0.1805	grain,ship	grain	0.2626
grain,wheat	rice	-0.1781	grain,wheat	rejects	0.3035
wheat,corn	exporters	-0.1771	grain,corn	maize	0.6120
wheat,corn	april	-0.1764	grain,corn	corn	0.7340
grain,corn	wheat	-0.1700	grain,wheat	wheat	0.9399

Table 12: *The highest and lowest factor values learned by CMLT using data ReutSM. Larger clique potentials influence the probabilitis for each combination more than smaller clique potentials. Thus, for instance, a combination involving labels* GRAIN *and* CORN *will have a higher score if the document has the terms* corn *or* maize*, but the classification scores are punished for documents having the noise term* blah. *The low parameter values corresponding to* $\langle$WHEAT, CORN, april$\rangle$ *and* $\langle$GRAIN, CORN, wheat$\rangle$ *suggest that the classification performance for smaller datasets such as these might be better if three labels and one term represent a clique potential, as well as two labels and one term.*

	HD-small			HD-big		
	Supported inference					
metric	Binary	CML	CMLT	Binary	CML	CMLT
macro-F1	0.585	0.625	0.620	0.647	0.676	0.662
micro-F1	0.614	0.645	0.644	0.684	0.699	0.698
subset accuracy	0.410	0.553	0.571	0.491	0.589	0.601
	Binary pruning					
metric	Binary	CML	CMLT	Binary	CML	CMLT
macro-F1	0.585	0.644	0.644	0.647	0.678	0.678
micro-F1	0.614	0.662	0.662	0.684	0.702	0.702
subset accuracy	0.410	0.570	0.570	0.491	0.584	0.584

Table 13: *F1 micro and macro averages and subset accuracy of the three classifiers on the OHSU-Med Heart Disease subsets, using 50% of features and with a binary pruning threshold of 70%. The multi-label models improve subset accuracy by as much as 53%.*

OhsuMed, pruned test instances						
metric	Binary	CML	CMLT	Binary	CML	CMLT
	HD-small			HD-big		
Binary Pruning, 50% features						
macro-F1	0.241	0.224	0.224	0.323	0.294	0.294
micro-F1	0.256	0.222	0.222	0.335	0.291	0.291
Supported, 50% features						
macro-F1	0.605	0.382	0.322	0.659	0.520	0.420
micro-F1	0.633	0.485	0.456	0.696	0.467	0.467

Table 14: *Performance of the three maximum entropy techniques on the heart disease pruned instances, using threshold 0.7 for binary pruning. For both supported inference and binary pruning, binary classification results in better performance on pruned instances, as with ReutAll, than the CML and CMLT models.*

HD-small				HD-big		
	Binary	CML	CMLT	Binary	CML	CMLT
macro-average						
mean	0.554411	0.633321	0.633517349	0.649873	0.675575	0.677784733
stderr	0.012484	0.013435	1.18E-16	0.001522	0.002171	0
micro-average						
mean	0.576348	0.637003	0.65167711	0.683152	0.69849	0.702646958
stderr	0.011804	0.011399	0	0.000895	0.001748	3.93E-17
	subset accuracy					
mean	0.369772	0.546834	0.567993367	0.496049	0.586004	0.584498602
stderr	0.01091	0.013007	1.18E-16	0.002158	0.003233	0

Table 15: *Mean micro- and macro-averages and subset accuracy of the binary pruning CML and CMLT classifiers over several trials using 50% of features at threshold 0.7, on HD-small and HD-big. Within error, CML and CMLT reduce error in micro- and macro-averages by about 17%, and about 7%, on HD-small and HD-big, respectively, and similarly improve subset accuracy by as much as 53% and 18%.*

$\langle$OTHER, INFORMATION SPACE, $menu\rangle$ in CMLT is among the highest valued model parameters. This seems intuitively clear: the appearance of the term MENU in a document may cause the binary classifiers to believe that the document belongs to the categories PRESENTATION and CONTROL, but the belief that the document belongs to the class INFORMATION SPACE supports the belief that it also belongs to the class OTHER (moreso than PRESENTATION or REPRESENTATION).

label	binary	CML	CMLT
Other	0.7702	0.7602	0.7581
Presentation	0.5359	0.4669	0.4073
Representation	0.5263	0.5153	0.5183
Control	0.5459	0.5261	0.5201
Application Component	0.3638	0.4008	0.3834
Information Space	0.5924	0.5740	0.5554
subset accuracy	0.2380	0.4172	0.3957

Table 16: *F1 measure for each label and subset accuracy of the three maximum entropy techniques for the usability data. Note that the character of the usability corpus and label taxonomy obviates binary pruning.*

Moreover, the usability corpus label taxonomy requires that each document receive exactly two labelings, one from *Interface Aspect* and one from *Interface Element*. The multi-label classifiers enforce this constraint implicitly, assigning a classification score for each document only for each of the nine subsets of labels. However, the binary classifiers may believe that a document belongs to any number of classes, so the binary classification scheme is likely to have poorer subset accuracy and poorer F1 than the multi-label schemes when the label taxonomy is structured, particularly with noisy data. However, independent (triary) maximum entropy classifiers for each label dimension still yielded about 12% poorer subset accuracy than the multi-label classifiers, though the triary classifiers tended to have commensurate F1 scores per label as the multi-label counterparts.

6 Related Work

The single-label maximum entropy algorithm is similar to the single-label maximum entropy algorithm described in [Nigam, Lafferty, and McCallum, 1999] which used improved iterative scaling to find the global maximum.

Aside from training independent binary classifiers to use in classifying a document as well as in ranking classes, as previously discussed, a less common approach to multi-label classification is to train a model that treats each subset of labels as a separate label. This approach can be cumbersome, however, since the number of such subsets is exponential in the number of labels. For semantic scene classification, [Boutell et al., 2003] train a classifier for each label c using all single-label documents and only the multi-label documents labeled c. Given d, each classifier reports a score for its respective label, and a similar ranking system is used to determine the multi-labeling. This approach indirectly leverages label co-occurrences, since each classifier is trained using a slightly different subset and partitioning of the training data. However it does not represent the co-occurrence patterns in one model, and does not repersent label-cooccurrences with respect to features. Furthermore this approach is subject to giving too much weight to examples with multiple labels, hence requires some method for reducing overfitting.

Bayesian approaches to multi-label text classification have been attempted using Expectation Maximization to train a mixture model [McCallum, 1999] that maintains a mixture component for each class, and learns the distribution over mixture weights as well as the distribution of words for each mixture component. Compared to other methods, it is most similar to our approach because it employs relationships

between classes in classifying an unseen document: the *terms* of each document are produced by a mixture of word distributions for each class. Ueda and Saito [Udea and Saito, 2004] take a similar approach in that each word in each category is generated from a multinomial distribution over vocabulary words. Both of these approaches are generative, and both leverage information about multiple class memberships for a given document implicitly by learning which classes generate which words, and neither experiment uses information gain in selecting features.

7 Conclusions

Multi-label classification is a task of growing importance in text domains as well as in other domains, and in most cases the classes are not independent. The maximum entropy approaches presented in this paper, in classifying a text document into a given class, leverage information about the document membership in other classes. Previous methods accomplish this indirectly, if at all, while the models described in this paper capture these relationships directly by maintaining factors that capture label co-occurrence patterns, and including these factors in the optimization. On corpora of varying sizes, label taxonomies and noise levels, in F1-scores and subset accuracy the multi-label models outperform the conventional multi-label classification methods, for which performance depends on independence of classes, simplicity of label taxonomy and sparsity of multi-labelings. Furthermore the multi-label models presented here outperform the conventional methods even when only a few documents have multiple labels, and perform comparably to best reported results on certain corpor. Experiments involving random test-train splits of various corpora suggest that the multi-label methods presented here perform better than their binary counterparts, barring error over trials.

8 Future Directions

CML and CMLT capture label co-occurrences in document labelings by capturing factors involving *pairs* of labels. A more complex corpus could conceivably maintain dependencies among triplets of labels, such that pairwise dependencies between labels in the triplet occur as well as a factor involving the simultaneous co-occurrence of all three labels. Space and time complexity challenge such directions, because this would significantly increase the number of model parameters, particularly for CMLT. Another direction would be to incorporate schemes for learning which factors to include and which factors to exclude.

Labels in Reuters-21578 tend to cluster with respect to multi-label documents. For instance, a document belonging to GRAIN is more likely to belong to WHEAT than to CRUDE. Although CML and CMLT learn this relationship, the models could be simplified or made less sensitive to noise by capturing the cluster more directly.

A straightforward solution is to preprocess the training data to capture the flavor of the label distribution, and use that information to determine how to count certain cliques. For example, CMLT could maintain parameter values only for label pairs that co-occur more than k times in the training data, for some threshold value k. This technique would improve classifier performance on large corpora such as Reuters-21578, where labels tend to cluster in their co-occurrence patterns. As an alternative to ignoring the less common co-occurrences, CMLT could selectively consider the potentials of the features for these label pairs. For instance, a document membership in two classes that otherwise co-occur infrequently is likely to depend on only a small number of words in the document.

Multi-labelings in the Reuters-21578 corpus that associate otherwise distinct clusters tend to involve articles whose subject is indirectly about some latent new topic. Thus, rather than ignoring the document or the newly associated cliques, the algorithm may give them special consideration.

The modals considered in this paper assume that each document belongs to at least one category. For Reuters-21578 however 47% of documents belong to no topical category. As it would be ideal to train the

models on unlabeled documents as well, it would also be ideal to allow the classifier to declare a document "unclassified". A proportion of such unlabeled documents may be treated as if they belong to a large *unclassified* class. If the documents truly do not belong to any of the existing categories, then training using those documents adds extra noise. A more complex scheme that would make the model less sensitive to noise incorporates a clustering heuristic in initializing maximum entropy constraints, allowing multiple related *unclassified* classes to exist based on clusters of similar documents. The success of this approach depends on the characterization that that the unlabeled documents among reasonably few clusters of similar documents, which may or may not be a likely characterization. It also depends on the degree to which unlabeled documents are not related to labeled documents.

Another improvement to the models involves more accurately representing label taxonomies which have a more complex structure than the simple and two-dimensional taxonomies described in this report, such as with hierarchical, multi-dimensional, or composite taxonomies, and more generally label taxonomies for which dependencies between labels comprise a graphical model or belief network.

Rsearch related to multi-label classification involves automatically annotating biomedical documents (or abstracts) with lists of genes that are mentioned within them. This is a difficult problem related to multi-label classification because each gene may have several synonyms, and a synonym may refer to several different genes. One future direction is to test the models using schemes with varying noise characteristics, and experiment with different techniques for discerning and reducing the noise.

Ideally the maximum entropy models developed in this paper will apply to domains other than text classification, such as domains in which the features do not have uniform weight and type, as with semantic scene classification.

References

[Beeferman, et al., 1999] Doug Beeferman, Adam Berger, John Lafferty (1999): Statistical Models for Text Segmentation. *Machine Learning, 34(1-3):177-210.*

[Berger, et al., 1996] Adam Berger, Stephen Della Pietra, and Vincent Della Petra (1996): A Maximum Entropy Approach To Natrual Language Processing. *Computational Linguistics 22-1*. Also available as *IBM Research Report RC-19694*.

[Boutell et al., 2003] Matthew Boutell, Xipeng Shen, Jiebo Luo, Chris Brown (2003): Multi-Label Semantic Scene Classification. *Technical Report 813*.

[Bunescu and Mooney, 2004] Razvan Bunescu, Raymond Mooney (2004): Relational Markov Networks for Collective Information Extraction. *Proceedings of the Twenty-First International Conference on Machine Learning (ICML-2004).*

[Byrd, et al., 1994] Richard H. Byrd, Jorge Nocedal, Robert B. Schnabel (1994): Representations Of Quasi-Newton Matrices And Their Use In Limited Memory Methods. *Mathematical Programming, 1994, pp. 129-156*

[Chen and Ho, 2000] Hao Chen, Tin Kam Ho (2000): Evaluation of decision Forests on Text Categorization. *Proceedings of the 7th SPIE Conference on Document Recognition and Retrieval.*

[Chen and Rosenfeld, 1999] Stanley F. Chen and Ronald Rosenfeld (1999): A Gaussian prior for smoothing maximum entropy models. *Technical Report, CMU-CS 99-108, Carnegie Mellon University, 1999.*

[Chieu and Ng, 2002] Hai Leong Chieu, Hwee Tou Ng (2002): Named Entity Recognition: A Maximum Entropy Approach Using Global Information. *Technical Report.*

[Crammer and Singer, 2002] Koby Crammer, Yoram Singer(2002): A New Family of Online Algorithms for Category Ranking. *Proceedings of the 25rd Conference on Research and Development in Information Retrieval (SIGIR), 2002*.

[Ellisseef and Weston, 2002] Andre Ellisseef, Jason Weston (2002): A Kernel Method for Multi-labeled Classification. *Neural Information Processing Systems (NIPS) 2004*.

[Gao, et al., 2004] Sheng Gao, Wen Wu, Chin-Hui Lee, Tat-Seng Chua (2004): A MFoM Learning Approach to Robust Multiclass Multi-Label Text Categorization. *ICML 2004*.

[Hersh, et al., 1994] W. Hersh, C. Buckley, T.J. Leone, and D. Hickman (1994): Ohsumed: an interactive retrieval evaluation and new large text collection for research. *17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94), pages 192-201, 1994*.

[Joachims, 1998] Thorsten Joachims (1998): Text categorization with Support Vector Machines: Learning with many relevant features. *Machine Learning: ECML-98, Tenth European Conference on Machine Learning*.

[Kschischang, 2001] Frank R. Kschischang, Brendan J. Frey and Hans-Andrea Loeliger (2001): Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory 47:2, February, 498-519*.

[Lewis and Ringuette, 1996] David D. Lewis, Marc Ringuette (1994): Comparison of Two Learning Algorithms for Text Categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*.

[Lierre and Tadepalli, 1997] Ray Liere, Prasad Tadepalli (1997): . Active Learning with Committees for Text Categorization. *Proceedings of AAAI-97*.

[Manning and Schtze, 1999] Chris Manning, Hinrich Schtze (1999): Foundations of Statistical Natural Language Processing. *MIT Press. Cambridge, MA*.

[McCallum, 1999] Andrew McCallum (1999): Multi-Label Text Classification with a Mixture Model Trained by EM. *AAAI'99 Workshop on Text Learning*.

[McCallum, Frietag, and Pereira 2000] Andrew McCallum, Dayne Frietag, Fernando Pereira (2000): Maximum Entropy Markov Models for Information Extraction and Segmentation. *Proceedings of the 17th International COnference on Machine Learning*.

[Mitchell, 1997] Tom Mitchell (1997): Machine Learning. *McGraw-Hill Science/Engineering/Math*.

[Nigam, et al., 1999] Kamal Nigam, Andrew McCallum, Sabastian Thrun, Tom Mitchell (1999): Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*.

[Nigam, Lafferty, and McCallum, 1999] Kamal Nigam, John Lafferty, Andrew McCallum (1999): Using Maximum Entropy for Text Classification. *IJCAI-99 Workshop on Machine Learning for Information Filtering, pp. 61-67. 1999*.

[Ratnaparkhi, 1996] Adwait Ratnaparkhi (1996): A Maximum-Entropy Part-of-Speech Tagger. *Proceedings of the Empirical Methods in Natural Language Processing Conference, May 17-18, 1996*.

[Ratnaparkhi, 1997] Adawit Ratnaparkhi (1997): A Linear Observed Time Statistical Parser Based on Maximum Entropy Models. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*.

[Reynar and Ratnaparkhi, 1997] Jeffrey C. Reynar, Adwait Ratnaparkhi (1997): A Maximum Entropy Approach to Identifying Sentence Bounadries. .

[Rosenfeld, 1994] Ronald Rosenfeld (1994): Adaptive Statistical Language Modeling. *Ph.D. thesis, Carnegie Mellon University, 1994*.

[Schapire and Singer, 1999] Robert E. Schapire, Yoram Singer (2000): BoosTexer: a boosting based system for text categorization. *Machine Learning, 39(2/3):135-168*.

[Udea and Saito, 2004] Naonori Ueda, Kazumi Saito (2004): Parametric Mixture Models for Multi-Labeled Text. *The Transactions of IEICEJ (The Institute of Electronics, Informations and Communication Engineers of Japan), Vol.J87-D-II, No.3, pp.872–883, 2004*.

[Yang, 1998] Yiming Yang (1998): An Evaluation of Statistical Approaches to Text Categorization. *Journal of Information Retrieval*.

[Yang, 2002] Yiming Yang (2002): High-Performing feature selection for text classification. *Association for Computing Machinery, Conference on Information and Knowledge Management*.